INFO-F-202
Langages de programmation 2
Université libre de Bruxelles

Projet

Justin Dallant John Iacono Alexis Reynouard

1 Sommaire

Le but de ce projet est de coder un jeu inspiré de Sokoban, un puzzle game créé en 1981 par Hiroyuki Imabayashi. Dans ce jeu, on contrôle un personnage se déplaçant sur des cases carrées, et le but est de pousser des boîtes pour les emmener sur des cases cibles. Une illustration d'un niveau est diponible sur la page Wikipedia de Sokoban et vous pouvez tester le jeu en ligne, sur ce site web par exemple.

Vous pouvez réaliser le projet individuellement ou par groupe de deux. Si vous le complétez en groupe, les deux étudiants recevront des notes identiques (sauf si un étudiant ne se présente pas à la présentation orale).

Vous devez remettre :
- Un zip avec votre code, d'autres fichiers nécessaires (vous pouvez utiliser des images ou des fichiers texte) et un Makefile.
- Un document au format pdf expliquant votre démarche (voir ci-dessous)

Ces documents sont dûs pour le 12 janvier à 23h59 pour la première session. Les dates de la session d'été seront annoncées ultérieurement.

La semaine du 16 janvier, un entretien oral en présentiel sera programmé au cours duquel vous ferez une démonstration de votre programme et répondrez aux questions.

Pour la deuxième session, le projet sera identique et sera rendu en août. La date exacte sera annoncée une fois que le calendrier des examens de la deuxième session sera fixé. Vous pouvez passer l'examen écrit en première session et faire le projet en deuxième session (ou inversement) si vous le souhaitez.

2 Notation

La notation prendra notamment en compte les éléments suivants :
- tâches terminées,
- rapport écrit,
- qualité du code,
- présentation orale et questions,
- retard.

Votre note maximale sera de 10 si vous ne terminez que les tâches de bases et passera à 20 si vous terminez également 10 des 12 tâches additionnelles. À l'exception des tâches de bases qui doivent être réalisées en premier, vous avez le droit d'effectuer une tâche sans avoir effectué les tâches précédentes. La qualité de votre code, votre rapport écrit, votre présentation

orale et vos réponses aux questions à l'oral détermineront votre note par rapport au maximum déterminé par le nombre de tâches que vous effectuez.

3 Tâches

Les cinq tâches de base.

Les tâches de bases consistent en la réalisation d'un niveau fonctionnel de Sokoban, avec :
- un personnage contrôlé au clavier qui peut se déplacer dans 4 direction sur un plateau de cases,
- des murs infranchissables,
- des éléments (boîtes dans le jeu original) que le personnage peut pousser,
- des cases cibles sur lesquels le joueur doit mettre les boîtes,
- un moyen de réinitialiser le niveau, que ce soit un bouton à cliquer ou une touche du clavier.

Quand le joueur a mis les boîtes sur les cases cibles, quelque chose doit se passer pour le notifier de son succès (que ce soit un texte ou une image qui apparaît, le niveau qui recommence, le passage au niveau suivant... à vous de voir).

Chacun des cinq éléments listés (personnage, murs, boîtes et cibles fonctionnelles) constitue une tâche.

Les tâches additionnelles.

Une fois les cinq tâches de base réalisées, vous pouvez vous atteler aux tâches suivantes.

1. **Animation des éléments en cours de déplacement.** Lorsque le personnage ou les boîtes se déplacent, animez ce déplacement (c'est à dire que les éléments ne se déplacent plus instantanément d'une case à la suivante mais "glissent" entre les deux).
2. **Cases de téléportation.** Ajoutez des cases qui, lorsque le personnage se place dessus, l'envoient vers une autre case correspondantes (et dans l'autre sens également) si aucune boîte ne la couvre.
3. **Boites de couleur.** Ajoutez des boîtes et des cases cibles colorées, de manière à ce que le niveau ne soit résolu que lorsque les boites de couleurs sont sur des cibles de la même couleur.
4. **Boîtes légères.** Ajoutez des boîtes légères. Le joueur doit pouvoir pousser plusieurs boîtes légères en même temps si elles se trouvent l'une derrière l'autre.
5. **Compteur de pas.** Ajoutez un élément sur l'écran, qui indique au joueur combien de pas il a effectué jusqu'à présent sur le niveau.
6. **Meilleur score de pas.** Affichez quelque part le score de pas le plus bas obtenu par le joueur pour chaque niveau. Ce score doit être enregistré et persister à la fermeture du programme. Il faut aussi pouvoir le réinitialiser.
7. **Écran d'accueil.** Lorsque vous démarrez le jeu, un écran d'accueil apparaît pendant une seconde avant que l'écran principal du jeu ne s'affiche. L'écran d'accueil doit inclure votre nom.
8. **Niveaux et sélection de niveau.** Implémentez différents niveaux que vous enregistrez dans un ou plusieurs fichiers. Vous devez y retenir l'emplacement des murs, des différents élements du jeu implémentés (boîtes colorées ou non, teleporteurs, cibles) et la position de départ du joueur. Il vous faut ensuite un écran de sélection de niveau et un moyen d'y accéder depuis le menu principal. L'écran de sélection de niveau peut être simple.

9. **Limite de pas.** Ajoutez une limite de pas maximum sur certains niveaux (cette tâche nécessite donc d'avoir fait plusieurs niveaux). Une victoire ne doit être déclenchée que si le joueur a gagné avec au plus le maximum de pas demandé, sinon une défaite est affichée. Le nombre de pas restants doit être affiché.
10. **Déplacement automatique à la souris.** Ajoutez la possibilité de cliquer sur une case à la souris pour que le personnage se déplace vers cette case s'il y a un chemin pour y parvenir sans déplacer de boites.
11. **Éditeur de niveau.** Ajoutez la possibilité de pouvoir modifier un niveau de manière interactive au lieu d'y jouer. Vous devriez pouvoir enregistrer les modifications et ajouter de nouveaux niveaux.
12. **Détection d'échec.** Une boîte qui se trouve dans un coin formé par des murs est bloquée : le joueur ne peut plus la déplacer. Une boîte qui se trouve dans un coin formé par des boîtes bloquées et/ou des murs et aussi bloquée. Si toutes les boîtes sont bloquées alors un message d'échec apparaît avec un bouton pour recommencer la partie.

4 Qualité du code

Votre score de qualité de code sera basé sur la façon dont vous codez en utilisant les principes de la programmation orientée objet.

- Tout doit être décomposé en classes et en fichiers.
- Aucune variable globale.
- Pas de logique répétée.
- Commentaires et documentation du code.
- Le code des méthodes individuelles doit en général être assez court.
- Utilisation correcte des constructeurs et des destructeurs.
- N'utilisez aucune des classes compliquées de fltk. Utilisez simplement les fonctions de dessin, placées dans des classes comme nous l'avons fait dans les laboratoires.
- Utilisation du modèle de conception MVC (couvert dans l'avant-dernière leçon).

Les classes de dessin de base qui appellent les fonctions fltk doivent être séparées de votre code spécifique à Sokoban.

5 Questions orales

Vous aurez une présentation orale pendant laquelle il vous sera demandé de faire une démonstration de votre programme et de votre code. Durant cette présentation nous vous poserons quelques questions à ce sujet. Nous vous demanderons peut-être d'apporter une modification mineure au code pour voir si vous comprenez votre propre code. Si vous avez réalisé le projet en équipe de deux, les deux membres de l'équipe doivent se présenter ensemble.

6 Rapport écrit

Vous devez inclure un rapport écrit au format PDF, d'au maximum 10 pages. Il n'y a pas de minimum de longueur, mais le rapport doit au minimum comprendre les points suivants

- Tâches. Indiquez les tâches que vous avez accomplies.

— Classes. Pour chaque classe, fournissez l'interface (pas le corps des méthodes) et décrivez brièvement le rôle de chaque classe et comment elle se rapporte aux autres classes.
— Logique du jeu. Supposons que je démarre le jeu, sélectionne le premier niveau (si la tâche 9 est terminée), attend 10 secondes, puis fais un coup. Décrivez en détail ce qui se passe dans votre code.
— Modèle-Vue-Contrôleur. Avez-vous utilisé ce modèle de conception ? Si c'est le cas, expliquez comment vos classes correspondent à ce modèle de conception. Sinon, expliquez vos choix.

Prenez garde au fait vous que ce rapport fera partie des critères qui influencerons votre note (bien que ce ne soit pas le critère principal), et pas un document optionnel à bâcler. En particulier, essayez de rendre celui-ci propre et clair pour le lecteur.

7 Retard

Vos points seront réduit de 1% pour chaque heure de retard.

8 Ce sur quoi vous n'êtes PAS noté

Vous êtes noté sur votre capacité à créer un programme volumineux, complexe et orienté objet. Vous n'êtes pas noté sur votre capacité à avoir un beau jeu. Normalement, un graphiste créerait de jolis éléments visuels, et vous êtes des informaticiens, pas des graphistes. L'aspect visuel devrait être assez bon pour que les différents éléments soient reconnaissables et que les animations soient raisonnables, mais aller au-delà n'aidera pas votre note.

Pour certains d'entre vous, l'écran des applications créées avec ftlk clignote lors de l'exécution de programme. Cela se produit le plus souvent si vous l'exécutez dans une machine virtuelle. Cela n'affectera pas votre note.

9 Plagiat

Nous devons aborder le problème du plagiat dans le code. Ce n'est pas facile, car il est tout à fait normal de couper et coller de petits morceaux de code. Cependant, il n'est pas acceptable que vous copiez toute la logique liée au déplacement du personnage et des boites d'autres étudiants ou du Web.

La règle suivante s'applique : si vous avez plus d'une seule ligne de code que vous avez obtenue ailleurs, vous devez inclure un commentaire avec l'endroit où vous l'avez obtenue (comme l'URL). Notons que vous êtes libre de partager toutes les classes que vous créez qui encapsulent un simple dessin fltk avec d'autres étudiants, tant que vous les publiez publiquement sur le canal de projet des équipes.

Toute violation de cette politique, même minime, peut entraîner un 0 sur le projet.

Nous ne ferons pas de distinction entre ceux qui ont fourni du code et ceux qui ont copié du code ; si deux groupes partagent le code sans attribution, les deux obtiendront un 0. Vous pouvez éviter cela en insérant simplement un commentaire !